3rd International Workshop on Genetic and Evolving Fuzzy Systems
Witten-Bommerholz, Germany, March 2008

Evolving Fuzzy Inferential Sensors for Process Industry

Plamen Angelov, *Senior Member IEEE*, Arthur Kordon, *Member IEEE,* and Xiaowei Zhou, *Student Member IEEE*

Abstract— **This paper describes an approach to design self-developing and self-tuning inferential soft sensors applicable to process industries. The proposal is for a Takagi-Sugeno-fuzzy system framework that has evolving (open structure) architecture, and an on-line (possibly real-time) learning algorithm. The proposed methodology is novel and it addresses the problems of self-development and self-calibration caused by drift in the data patterns due to changes in the operating regimes, catalysts ageing, industrial equipment wearing, contamination etc. The proposed computational technique is data-driven and parameter-free (it only requires a couple of parameters with clear meaning and suggested values). In this paper a case study of four problems of estimation of chemical properties is considered, however, the methodology has a much wider validity. The optimal inputs to the proposed evolving inferential sensor are determined *a priori* and off-line using a multi-objective genetic-programming-based optimization. Different on-line input selection techniques are under development. The methodology is validated on real data provided by The Dow Chemical Company, USA.**

I. INTRODUCTION

A. Background and state-of-the-art

INFERENTIAL sensors also known as soft sensors [1], are applied nowadays extensively in a range of industries, such as processing, chemical, petro-chemical, manufacturing, etc. One of the typical application of soft sensors is for process quality monitoring [2],[3]. The black-box-model-based inferential sensors applied currently [1],[4],[5] has big advantages over the conventional solutions that rely on laboratory tests and manual intervention in terms of overall process automation and costs. The most widespread methods that are used to design inferential sensors are principle component analysis (PCA) for reducing the input dimensions and correlation between raw data readings and partial least squares (PLS) to train the models [1]. Alternative techniques that are used for soft sensors design are neural networks (NN) [6], support vector machines [7], genetic programming [8]. The main problem of inferential sensors based on these techniques is caused by the fact that the real industrial processes are highly non-linear, non-stationary, they have different operating regimes, the environment and the industrial equipment, raw materials and catalysts are changing. This dynamically (often unpredictably) evolving environment leads to pre-trained and designed in off-line mode inferential sensors to have unacceptable drop in their performance and to require periodic and costly re-training, re-calibration and sometimes re-development. In this way, the life-cycle costs of these sensors become comparable or higher than that of laboratory tests. Another significant disadvantage of NN and other 'black box' techniques is their lack of interpretability and transparency, which is very important for human operators of these expensive industrial processes who sometimes rely on their experience and intuition.

This reality calls for the development of new techniques that are adaptive and able to react to the complex changes in the process such as wearing out or contamination of the equipment, quality alteration of raw materials, etc. Ideally, an intelligent inferential sensor will have an online (possibly in real-time) structural learning ability in response to the fundamental shifts in the process.

B. Modes of operation of inferential industrial sensors

In most industrial process, the inferential sensors are expected to work in one of the following three modes of operation. For cases with high degree of stationarity and low level of variability of the raw materials, catalysts, environment and the equipment inferential sensors with a pre-trained fixed structure and parameters may be satisfactory. For processes with frequent non-fundamental changes, sensors are required to continuously adapt to these changes online. Such inferential sensors can have a fixed structure but require the ability to automatically re-tune their parameters in response to the changes. These sensors will be called adaptive or self-adaptive (if adaptation is automatic and on-line, but does not concern the structure of the inferential sensor) and self-tuning inferential sensors. Finally, there is also a type of inferential sensors that will be called evolving sensors, eSensors which evolve their structure as well as adapt their parameters. They are especially important and needed when there is a significant shift in the data pattern, which changes the way the original process works. Mathematical models with fixed structures are not able to react with parameters tuning to such situations.

In order to automatically detect, learn and react to both fundamental and non-fundamental changes, specific novel techniques and methodologies are needed, which requires the soft sensor to detect shifts and drifts in the process and learn not only the parameters, but also to evolve its own structure in

Dr. P Angelov and Mr. X Zhou are with the Intelligent Systems Research Lab, Dept of Communication Systems, InfoLab21, Lancaster University, Lancaster, LA1 4WA, UK; phone +44 (1524) 510391; e-mail: p.angelov@lancaster.ac.uk
Dr. A Kordon is with the Data Mining and Modelling Group, Corporate Work Processes & Six Sigma Centre, The Dow Chemical Company, Freeport, TX, USA; phone +1 (979) 238 5149; e-mail: akkordon@dow.com

978-1-4244-1613-4/08/$25.00 ©2008IEEE

online mode. One framework that can be used to design automatically such *eSensors* is the recently developed evolving Takagi-Sugeno (eTS) fuzzy systems [9],[10],[11].

C. *Illustrative case studies*

In some industrial processes, the real measurements are not available continuously or as often as the output is required. The real measurements needed for training the soft sensor might be collected over several sampling time intervals, which for different industrial processes may constitute several seconds, minutes, hours, several days or even more (as is the case with oil refinery or waste water treatment processes). The training samples may also come in batches. In such cases the options for designing an inferential sensor are; i) using filtering to estimate the output; ii) use unsupervised techniques; iii) periodically re-train in a batch mode the inferential sensor; iv) re-train the inferential sensor in on-line mode whenever the training data are available. In the latter case, during the periods in time when training data is not available, the inferential sensor will make predictions of the output based on the existing rule-base at the time. Ideally, an inferential sensor should be able to adapt and re-train 'on the fly', without interrupting the online estimation whenever new, fresh training data is available. This ability is critical when the inferential sensor is installed on a non-stop system, which does not allow offline re-training and most of the industrial installations in the chemical, bio- , and petro-chemical branches of industry are of this type.

Figure 1 illustrates several alternative cases for an example from a real chemical process which has a significant change in the operating conditions around sample 113. For example, the conventional mode of using inferential sensors in two clearly separated phases: i) off-line pre-training; ii) on-line use. This is illustrated for the same real data in Figure 1a).

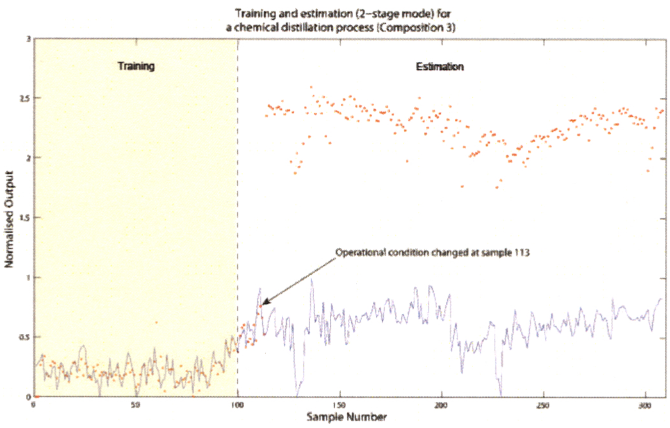

Fig. 1a) The typical mode of using inferential sensors: off-line pre-training (in this case for 100 time steps) and afterwards use in on-line mode with a fixed structure and parameters

One can see that due to the change in the operating conditions at sample 113 the predictions drastically deteriorate because the sensor (model) structure is ***fixed*** and does not have ability to evolve and to reflect the change in the data pattern by change in the rule-base. As a result a significant error is generated.

A possible solution would be to collect enough data and re-train the original inferential sensor in the same way as originally designed (in off line mode). The downsides of such an approach are; i) it increases considerably the cost of the development and maintenance of the sensor and the overall life-cycle costs; ii) the time of re-development and re-calibration may be significantly larger than the time interval of collection of next data sample(s); iii) such a mode leads to complete loss of previously collected information and knowledge.

Ideally, one would adapt gradually the existing inferential sensor to the newly collected data in on-line mode (possibly in real-time) by an automatic procedure that adapts the parameters but if necessary also upgrades or modifies the structure of the sensor. This is the approach that is followed in this paper. As shown in Figure 1b), the effect from the operating conditions change is dealt with in online re-training stage, without an interruption of the predictions.

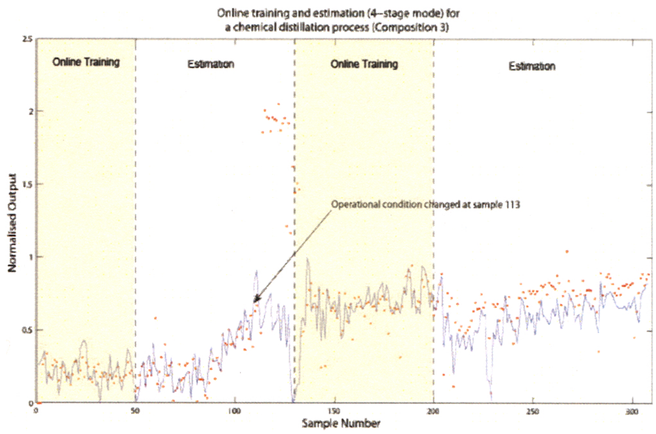

Fig. 1b) eSensor periodically re-trains and adapts both its structure and parameters to the change in the operating conditions of the industrial process.

Note that the proposed inferential sensor can also work in a fully online manner. When the feedback is available at each sampling time, the internal model of the inferential sensor can gradually evolve at every sampling interval.

II. EVOLVING INFERENTIAL SENSOR DESIGN

A. *The Concept of eTS*

The Evolving Fuzzy Rule-based System (eR) [9] introduced the approach of autonomous learning and evolving the structure of fuzzy rule-based systems together with its parameters. eTS is a special case of eR when the underlying structure of the model is of Takagi-Sugeno (TS) type and it self-develops from a data stream recursively [10,11]. The structure of eTS is a flexible and open set of TS fuzzy rules, which can grow, shrink, update '*on the fly*' according to the information (mainly density) brought by the new data. It is globally non-linear, and linguistically interpretable. In this way, the proposed eSensor (Figure 2) has embedded eTS that brings the ability to self-develop, self-calibrate, and self-maintain. Therefore, it is a promising tool that can suit the new demands from industry for fully autonomous inferential sensors which give interpretable models.

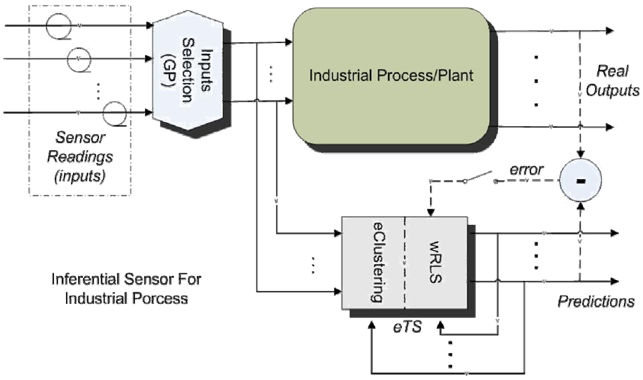

Fig. 2 Schematic representation of eSensor

B. Online Learning of eSensors

The learning method of eTS is based on two stages that

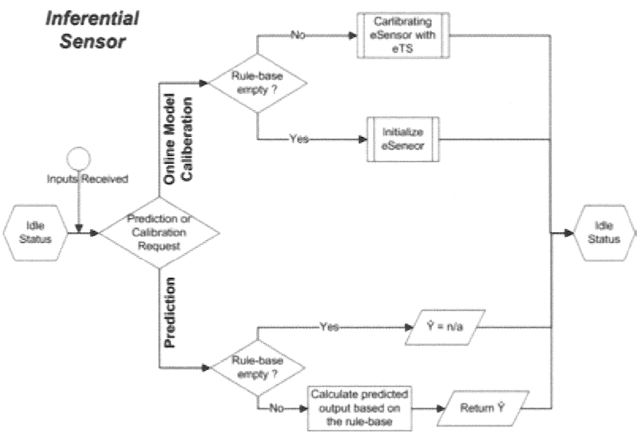

Fig. 3 Flow chart describing the work of eSensor

perform during a single time interval: i) data partitioning through evolving clustering; ii) fuzzily weighted recursive least squares estimation of the parameters of the consequents. The learning procedure is briefly described in the Appendix and is illustrated in Figure 3. More details about eTS learning can be found in [10,11].

This eTS-based evolving inferential sensor has the following specific features that separate it from existing sensors used in the process industries:

a) It has an ***evolving*** (open, flexible) structure and the evolution can start from scratch;
b) Due to its recursive calculations, very low memory is required for the calculations;
c) Due to its very low computational costs it can respond very quickly and is suitable for real-time applications;
d) Benefiting from the multiple local Kalman filters, it provides high prediction rates;
e) It has MIMO structure [13], and has the ability to model multiple outputs in a more efficient way.
f) By online monitoring quality of the clusters and fuzzy rules it is possible to automatically detect *shifts* in the data pattern that reflect different operating regimes.

C. Monitoring quality of the fuzzy model online

In order to analyze quality of the inferential sensor, several aggregated variables have been defined and monitored online. This includes *age* of the cluster/fuzzy set which is defined [14] as:

$$Age^i = k - \frac{1}{n_k^i}\sum_{l=1}^{n_k^i} k_l$$

where k_l is the time index when a data sample is read.

Age of i^{th} rule is updated by adding *1*, indicating that it gets *older*, unless new sample activates this rule. Therefore, the range of *age* is *[0,k]*. In evolving modeling, it is vitally important to adapt to the *shift* in the data pattern. Detecting the concept *shift* helps verifying the structure of the evolving fuzzy rule-base online. *Old* fuzzy rules do not reflect the up-to-date real information coming from 'hard' sensors, thus need to be disabled or removed. In Figure 4 real data from propylene production is used. One can see that fuzzy rule *1* is intensively updated around sample *1450*, which reflects a change in the operating mode that occurs in the real plant. Similarly, around sample *2700* rule *4* is activated (its age drops) while rule *1* gets older due to less intensive or no update. These two significant shifts in the data pattern can be automatically detected by analyzing online the first and second derivative of *age* in time.

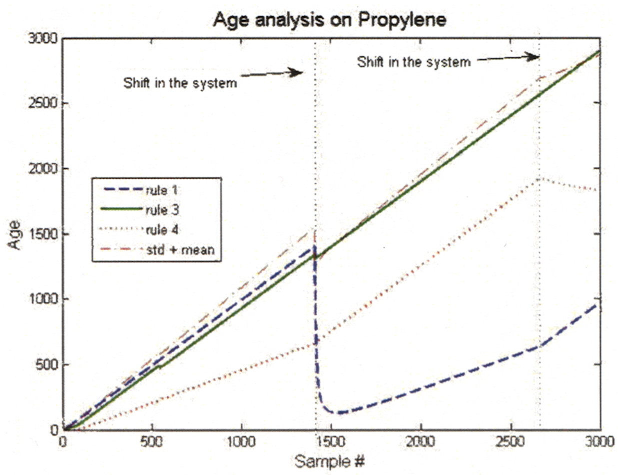

Fig. 4 Propylene production data – significant shifts in the data pattern take place around samples 1450 and 2700 which lead to quick update of fuzzy rule *1* and slower update of rule *4*.

III. SENSOR INPUTS SELECTION

In order to enable the inferential sensor, which do not rely on expert knowledge, to be completely autonomous one critical step is to select the most informative inputs from all the available 'hard' (conventional) sensors. Removing noisy or irrelevant inputs that usually lead to a drop in modeling precision improves the estimation quality and efficiency of the proposed inferential sensors.

Conventionally, the model input selection is carried out in off-line mode as a part of the design of the inferential sensor [2,15]. There are a number of widely applied techniques such as PCA [1], Genetic Programming (GP) [2,15,17] etc. In the proposed case study a multi-objective GP is used for the input selection task due to its capability for symbolic regression [16]. GP simulates the natural evolution of a number of potential candidates to be selected according to the objective function [8]. As a result, the selected candidates are normally best suited to fit the objective function(s). This approach has the advantage that no *prior* knowledge is required abut the model structure and the result is based on explicit objective function(s). Pareto-based multi-objective formulation is used in this particular application to compensate the better precision with the model complexity.

The input selection, based on GP relates the sensitivity of a given input variable to its fitness in the population of equations. The reasoning is that important input variables will be used in equations that have a relatively high fitness. The algorithm is described in [17] and the results of its application for feature selection in one of the case studies — Composition

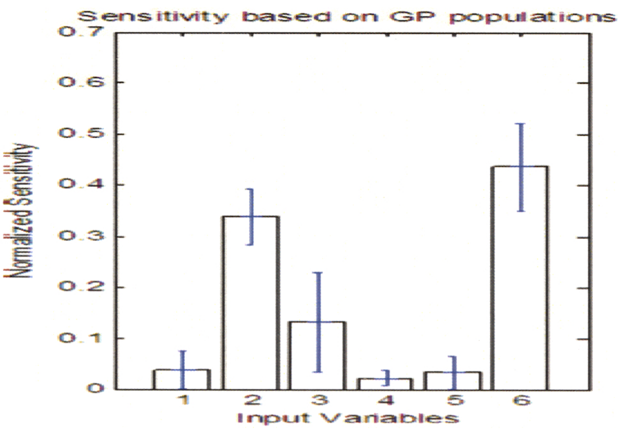

Fig. 5 Input selection for Composition 1 based on GP. Only two out of six input variables (x2 and x6) are with high nonlinear sensitivity to the output.

1, is shown in Figure 5.

The proposed multi-objective GP method generates many non-dominated solutions on the Pareto front, shown with encircled dots on Figure 6.

Usually the explored area for model selection is narrowed down to the section on the Pareto front with the biggest gain in accuracy for the smallest expressional complexity. An example of explored models area on the Pareto front for selecting the Composition 2 model is shown in Figure 6. The performance of the individual models is explored as well as the interpretability of the derived functional forms from physical consideration. The final model selection is done by the inferential sensor users. In some cases, several models with different inputs are selected to improve the robustness in case of input failure. The applied inferential sensor is an ensemble of models.

For a system which works on data stream which requires the sensor to perform in online mode, the input selection has to be done prior to the work of the system or alternative approaches must be developed that allow online inputs selection. This is a direction of current research and in this paper we assume that the inputs have been defined prior the on-line phase.

IV. CASE STUDIES AND EXPERIMENTAL RESULTS

Four problems from the chemical industry have been considered as case studies. The eTS-based *eSensor* has been applied for prediction of the properties of three compositions and propylene in a simulated online mode. Each of the datasets includes different impact due to the changes of the operating regime of the process, which bring challenges to the

TABLE I
GENERAL INFORMATION FOR THE 4 DATA SETS

	Comp1	Comp2	Comp3	Propylene
All measured inputs	6	47	47	22
Selected Inputs by GP	2	2	7	2
No of Samples	309	308	308	3000
Noise	Yes	no	Yes	Yes
Operating regime change at sample	127	113	113	Broad range of operations

structure of the inferential sensor. The four test cases include a number of other challenges, such as noise in the data, a large number of initial variables, etc. These problems cover a wide range of real issues in the industry when an inferential sensor is to be developed and applied. The proposed new inferential sensors, *eSensor* proved to be capable of being an advanced replacement of the existing less flexible solutions.

The first case, called in this paper 'Composition *1*', is to model the product composition in a distillation tower. The process data is retrieved from *6* physical ('hard') sensors used as inputs to the inferential sensor applying hourly averages for every *8* hours. The product composition (real output) is estimated by a laboratory analysis. The estimation of the product composition contains noise due to the nature of the analysis. A significant operating condition change has taken place after sample *127*.

The second case, called 'Composition *2*' concerns product composition in the bottom of the distillation tower. A list of *47* related variables are initially included as the inputs, some of them are very loosely related to the product composition. Similarly, laboratory analysis has been used to obtain the real output, which is less noisy than the output for the other *3* datasets. There is a significant operation change around data sample *113*.

The third case, Composition *3*, is very similar to the previous case (Composition *2*). The only difference is that the level of noise in this problem is much higher.

The fourth case concerns the Propylene that is in the top of the distillation tower. *22* different physical ('hard') sensors and respectively inputs for the inferential sensor are measured. The data for this, fourth case contains *3000* data points measured every 15 minutes using gas chromatography. They cover a very broad range of operating conditions.

GP was applied off-line prior to the design of the evolving inferential sensor with the aim to select the optimal subset of inputs based on Pareto front [15]. Table I describes the experimental set up in brief.

Two tests of the eTS-based inferential sensor were carried out with each of the four test cases; i) using all inputs; ii) using pre-selected (by GP) optimal sub-subset of the inputs.

eSensor starts to learn and generate its fuzzy rule-base from the first data sample it reads. The prediction starts when the rule-base is initialized straight after the first sample is read. After that, eSensor evolves the fuzzy rule-base structure (Figure 6 illustrates this for Composition 3) on a sample by sample basis and adapts the parameters of each rule in the rule-base online. In this way, the inferential sensor continuously adapts and self-calibrates.

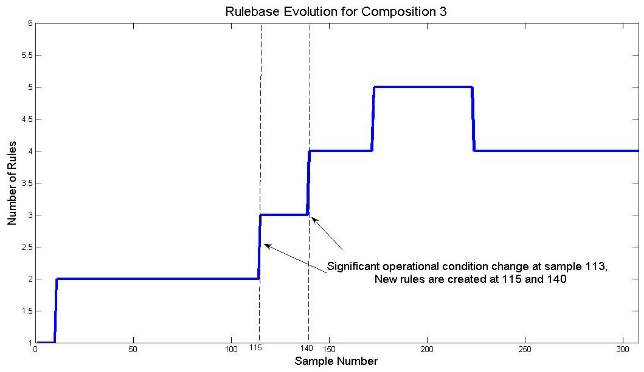

Fig. 7 Evolution of the fuzzy rule base of eSensor for Composition 3

The fuzzy rule-base that was automatically generated for the propylene after all data samples has been read is shown in Figure 7. Note that it evolved 'from scratch' by generating rules one by one (see Figure 6 for the fuzzy rule evolution for the case of Composition 3) where $\bar{y}, \bar{x}$ denote the normalized inputs and outputs.

RULE-BASE for Propylene using selected inputs:

R_1: ***IF*** *(x_1 is around 24.6)* ***AND*** *(x_2 is around 26.3)* ***THEN*** *(* $\bar{y} = -0.039 + \bar{x}_1 - 0.324\bar{x}_2$ *)*

R_2: ***IF*** *(x_1 is around 39.0)* ***AND*** *(x_2 is around 43.5)* ***THEN*** *(* $\bar{y} = -0.615 + 4.77\bar{x}_1 - 0.340\bar{x}_2$ *)*

R_3: ***IF*** *(x_1 is around 46.2)* ***AND*** *(x_2 is around 49.5)* ***THEN*** *(* $\bar{y} = -0.679 + 1.090\bar{x}_1 + 0.450\bar{x}_2$ *)*

R_4: ***IF*** *(x_1 is around 45.9)* ***AND*** *(x_2 is around 49.9)* ***THEN*** *(* $\bar{y} = -1.340 + 5.570\bar{x}_1 - 3.320\bar{x}_2$ *)*

R_5: ***IF*** *(x_1 is around 36.2)* ***AND*** *(x_2 is around 43.5)* ***THEN*** *(* $\bar{y} = -0.002 + 0.320\bar{x}_1 - 0.065\bar{x}_2$ *)*

R_6 ***IF*** *(x_1 is around 31.6)* ***AND*** *(x_2 is around 38.7)* ***THEN*** *(* $\bar{y} = -0.007 + 0.366\bar{x}_1 - 0.129\bar{x}_2$ *)*

The predictions made by *eSensor* for the case of Composition 3 are plotted against real data in Figure 8. Note that this eTS based inferential sensor may also start updating an existing fuzzy rule-base that might contain expert knowledge.

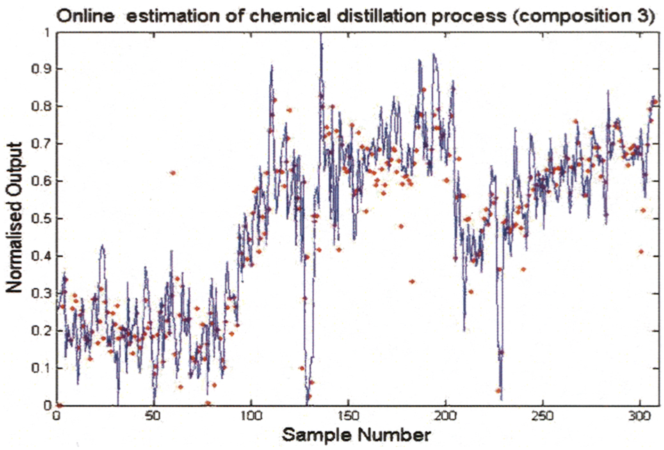

Fig. 8 Prediction of eSensor versus real data for composition 3

The numerical results of the precision of eSensor are shown in Tables II and III (NDEI: non-dimensional error index; VAF: variance accounted for with ideal value 100%). Both tests (using all inputs and using selected smaller subset of inputs) give a good level of accuracy for all four cases. The number of fuzzy rules generated is also small and the fuzzy sets are linguistically interpretable. Good results demonstrate that eSensor can successfully address the impact of changing operating conditions by automatic evolution.

From Table III, one can see that when using fewer inputs the overall accuracy of eSensor can even be improved. This is due to the fact that less important and correlated inputs have been removed. It is also interesting to note that the number of fuzzy rules that have been generated using a smaller optimal subset of inputs is smaller. More importantly, with less inputs used, the antecedent part of the fuzzy rules become more succinct and thus enables the interpretably of the rule-base by the operators and later laboratory analysis.

TABLE II
RESULT FOR EXPERIMENT USING ALL MEASURED INPUTS

	Comp1	Comp2	Comp3	Propylene
NDEI	0.301	0.327	0.423	0.187
VAF, %	90.98	89.41	82.07	96.51
# Rules	4	4	5	6
# Features	6	47	47	23

V. CONCLUSIONS

In this paper an eTS-fuzzy-system-based inferential sensor, eSensor was described and applied to four test problems from chemical process industry. The proposed inferential sensor is

TABLE III
RESULT FOR EXPERIMENT USING SELECTED INPUTS

	Comp1	Comp2	Comp3	Propylene
NDEI	0.333	0.254	0.419	0.141
VAF, %	89.22	95.79	82.91	98.00
# Rules	2	3	4	6
# Features	2	2	7	2

self-developing and self-calibrating. It proved to be efficient to automatically detect shifts in the data pattern and to flexibly evolve its structure of fuzzy rules that is also linguistically interpretable. It is an effective alternative to the currently existing inferential ('soft') sensors that offers lower maintenance and life-cycle costs. GP was used to pre-select the optimal small number of inputs. The future research is directed towards development and incorporation of online inputs selection in the process of the evolving inferential sensor.

VI. APPENDIX

A brief outline of the basic procedure used by eSensor, eTS is given here. More details can be found in [10,11]. eSensor develops and uses a fuzzy rule-based TS fuzzy model (in general it can be MIMO) as the one described in Figure 7. One can use a concise mathematical expression in a vector form:

$$y = [\lambda^1 \bar{x}^T \quad \lambda^2 \bar{x}^T \quad \cdots \quad \lambda^N \bar{x}^T] \cdot \begin{bmatrix} (\theta^1)^T \\ (\theta^2)^T \\ \cdots \\ (\theta^N)^T \end{bmatrix} \tag{A1}$$

where θ^i denotes the vector of consequent parameter of the i^{th} fuzzy rule; λ^i, *i=[1,N]*; is the normalized firing level of i^{th} local fuzzy rule; $\lambda^i = \frac{\tau^i}{\sum_{j=1}^{N}\tau_j}$; *N* is the number of fuzzy rules; $\tau^i = \mathop{\mathrm{T}}_{j=1}^{n} \mu_j^i(x_j)$ is the overall firing level of i^{th} rule which can be determined by a t-norm (usually product operator); $\mu^i = e^{-\sum_{j=1}^{n}\frac{(x_j - x_j^{i*})^2}{2(\sigma_j^i)^2}}$ is the membership function of the i^{th} rule usually described by a Gaussian; x_j^{i*} is the focal point of i^{th} fuzzy rule which defines its antecedent part; σ_j^i represents the spread of Gaussian function for j^{th} fuzzy sets, which can be learned from the data distributions [11] by $(\sigma_{jk}^i)^2 = \rho(\sigma_{j(k-1)}^i)^2 + (1-\rho)\frac{1}{n_k^i}\sum_{l=1}^{n_k^i}\left\|z^{i*} - z_l\right\|_j^2$ $\sigma_{j1}^i = 1$; ρ is the learning rate (suggested value 0.5); n_k^i denotes the *support* of the i^{th} rule, which is calculated on the numbers assigned to the rule based on the distance to the focal point.

REFERENCES

[1] L. Fortuna, S. Graziani, A. Rizzo, M. G. Xibilia, Soft Sensor for Monitoring and Control of Industrial Processes, In Advances in Industrial Control Series (M. J. Grimble, M.A. Johnson Eds.), Springer Verlag: Berlin, Germany, 2007.

[2] A. Kordon and G. Smits, *Soft Sensor Development Using Genetic Programming*, Proceedings of GECCO'2001, pp. 1346 – 1351, 2001.

[3] J. J. Macias-Hernandez, P. Angelov, X. Zhou, Soft Sensor for Crude Oil Distillation Side Streams using Takagi Sugeno Evolving Fuzzy Models, Proc. *2007 IEEE Intern. Conf. on Systems, Man, and Cybernetics*, 7-10 Oct., 2007, Montreal, Canada, ISBN 1-4244-0991-8/07, pp.3305-3310.

[4] W. Yan, H. Shao, X. Wang, Soft sensing modeling based on support vector machine and Bayesian model selection, Computers and Chemical Engineering, Vol. 28, pp.1489-1498.

[5] L. Chen, O. Bernard, G. Bastin, P. Angelov, Hybrid Modelling of Biotechnological Processes using Neural Networks, *Control Engineering Practice*, vol.8, No7, 2000, pp.821-827.

[6] R. Rallo, J. Ferre-Gine, A. Arena, F. Giralt, Neural virtual sensor for the inferential prediction of product quality from process variables, Computers and Chemical Engineering, Vol. 26, pp.1735-1754.

[7] V. N. Vapnik, *The Statistical Learning Theory*, Springer, 1998.

[8] J. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press, Cambridge, MA, 1992

[9] P. Angelov, *Evolving Rule-based Models: A Tool for Design of Flexible Adaptive Systems*. Heidelberg, Germany: Springer-Verlag, 2002.

[10] P. Angelov, D. Filev, "An approach to on-line identification of evolving Takagi-Sugeno models", *IEEE Trans. on Systems, Man and Cybernetics, part B,* vol.34, No1, pp. 484-498, 2004.

[11] P. Angelov, X. Zhou, Evolving fuzzy systems from data streams in real-time, *Proc. 2006 International Symposium on Evolving Fuzzy Systems*, 7-9 September, 2006, Ambleside, Lake District, UK, IEEE Press, pp.29-35, ISBN 0-7803-9719-3.

[12] R. Babuska, Fuzzy Modeling for Control, PhD thesis, Technical University of Delft, The Netherlands, 1997.

[13] P. Angelov, C. Xydeas, D. Filev, On-line identification of MIMO evolving Takagi-Sugeno fuzzy models, Intern. Joint Conf. on Neural Networks and Intern. Conf. on Fuzzy Systems, IJCNN-FUZZ-IEEE, Budapest, Hungary, 25-29 July, 2004, 55-60, ISBN 0-7803-8354-0.

[14] P. Angelov, X. Zhou, E. Lughofer, D. Filev, Architectures of Evolving Fuzzy Rule-based Classifiers, Proc. 2007 IEEE International Conference on Systems, Man, and Cybernetics, 7-10 October, 2007, Montreal, Canada, ISBN 1-4244-0991-8/07, pp.2050-2055.

[15] A. Kordon, G. Smits, A. Kalos, and E. Jordaan, Robust Soft Sensor Development Using Genetic Programming, In *Nature-Inspired Methods in Chemometrics*,(R. Leardi-Ed.),pp. 69–108, Elsevier, Amsterdam, Holland, 2003.

[16] G. Smits, M. Kotanchek, Pareto-Front Exploitation Symbolic Regression, *In Genetic Programming Theory and Practice II, U.M O'Reilly, T. Yu, R. Riolo, B. Worzel (Eds)*, Springer, NY, USA, pp. 283-300, 2004.

[17] G. Smits, A. Kordon, E. Jordaan . C. Vladislavleva, and M. Kotanchek, Variable Selection in Industrial Data Sets Using Pareto Genetic Programming, In: Yu T., Riolo, R. and Worzel, B. (eds): *Genetic Programming Theory and Practice III*. Springer, NY, pp. 79 - 92, 2006.